

Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX 76019



Scalable Holistic Analysis of Multi-Source Data-Intensive Problems Using Multilayered Networks

Abhishek Santra, Sanjukta Bhowmick and Sharma Chakravarthy

Technical Report

arXiv:1611.01546v1 [cs.DB] 4 Nov 2016

Scalable Holistic Analysis of Multi-Source, Data-Intensive Problems Using Multilayered Networks

Abhishek Santra^{*1}, Sanjukta Bhowmick^{†2}, and Sharma Chakravarthy^{‡1}

¹IT Lab, CSE Department, University of Texas at Arlington, Texas, USA

²Department of Computer Science, University of Nebraska at Omaha, Nebraska, USA

November 8, 2016

Abstract

Holistic analysis of many real-world problems are based on data collected from multiple sources contributing to some aspect of that problem. The word fusion has also been used in the literature for such problems involving disparate data types. Holistically understanding traffic patterns, causes of accidents, bombings, terrorist planning and many natural phenomenon such as storms, earthquakes fall into this category. Some may have real-time requirements and some may need to be analyzed after the fact (post-mortem or forensic analysis.) What is common for all these problems is that the amount and types of data associated with the event. Data may also be incomplete and trustworthiness of sources may also vary. Currently, manual and ad-hoc approaches are used in aggregating data in different ways for analyzing and understanding these problems.

In this paper, we approach this problem in a novel way using multilayered networks. We identify features of a central event and propose a network layer for each feature. This approach allows us to study the effect of each feature independently and its impact on the event. We also establish that the proposed approach allows us to compose these features in arbitrary ways (without loss of information) to analyze their combined effect. Additionally, formulation of relationships (e.g., distance measure for a single feature instead of several at the same time) is simpler. Further, computations can be done once on each layer in this approach and reused for mixing and matching the features for aggregate impacts and "what if" scenarios to understand the problem holistically. This has been demonstrated by recreating the communities for the AND-Composed network by using the communities of the individual layers.

Specifically, we propose a representation of disparate data as multilayered network, that can capture the inter-relation between the events and makes it easier to add new information to individual layers as they become available. Further, algorithms have been given for combining multiple layers in any arbitrary manner in order to facilitate the study of the combined effect of different sources. Finally, we present an elegant and low-cost method to combine analytical results from multiple layers, without recomputing the combined layers.

We therefore believe that techniques proposed here make an important contribution to the nascent yet fast growing area of data fusion.

^{*}abhishek.santra@mavs.uta.edu

[†]sbbhowmick@unomaha.edu

[‡]sharma@cse.uta.edu

1 Introduction

A critical aspect of big data analysis is identifying how different features, collectively or individually contribute to a central event. From natural phenomena such as storms, earthquakes, to traffic accidents, to premeditated crimes such as terrorist attacks, all events are multifaceted in nature. Multiple data sources capture different perspectives for each event. The central question in analyzing such multifaceted data is to study that how the individual and combinations of sources effect the events for a holistic understanding of the central event.

Motivation: As an example, consider the case of traffic accidents. Each traffic accident is tabulated with associated information (also termed features) such as the geographical location, the date, the time when it happened, the light and weather conditions at the time of occurrence, the number of casualties, the number, type and speed of vehicles, type of the locality (urban or rural), type of the road (one way, roundabout etc.) and the people involved in the accident. The associated data for these features is captured by various sources. Each feature or combination of features tells a different story about the same set of accidents. For example, two accidents even if they occurred during the same time of the day with similar light and weather conditions involving same type and number of vehicles, may lead to different number of casualties due to marked difference in speed at the roundabout.

Therefore, given such a database of traffic accidents, and associated features, if we can identify accidents that occurred primarily due to poor light, that occurred primarily due to bad weather, or those that occurred due to combination of both, then we can determine with respect to per accident location whether to have infrastructure to improve the lighting or to have warning signs due to bad weather or both. Such selective targeting of features, per event, is useful for responding to other problems that involve multiple features. For example, when a disease is treated by a cocktail of different drugs, physicians often manually tune the dosage of different drugs based on the patient's reactions.

Problem Formulation and Challenges: Given a dataset for a central event with multiple instances and associated features, the specific problem we want to address in this paper is to group these instances based on the features. To do so, we have to consider all possible subsets of the features, as every subset will bring out a distinct aspect of the central event. For each subset combination and event instance pair, we cluster event instances that occur at the same value of the feature combination. This is clearly a very computationally intensive task, because for n features, we will have 2^n possible subsets, leading to 2^n clustering problems. Furthermore, these clusterings will have to be recomputed each time new entries are added to the dataset.

In this paper, we present an elegant graph-theoretic technique using which we can efficiently cluster event instances based on different subsets of features.

Outline of Our Approach: Our approach to this problem is to represent the information as a multilayered network. Network analysis has become a very popular tool for analyzing systems of interrelated entities. The entities (here the event instances) are represented as vertices. Two vertices (event instances) are connected by an edge, currently unweighted and undirected, if the corresponding feature value between them is similar¹. Therefore instances that have similar features will be tightly connected together and form communities. In this paper, we will focus solely on community detection.

Because we have different features, we can create a separate network, each corresponding to a different feature. Such a set of networks, where the vertices are the same, but the connections between them vary, are collectively called *multilayered networks*.

¹It can be argued that weighted edges might provide a more faithful representation; however, here our goal is to simply connect two events that satisfy a certain level of similarity.

Representation of a multi source dataset as a multilayered network, provides several benefits. *First*, networks provide an elegant way of representing similar event instances on a per feature basis. Note that although the feature type might vary, from numeric, to nominal, to time, in each network they are canonically defined by edges. *Second*, it is relatively easy to combine the features. In most other scenarios, it is difficult to combine features having different types and domains of values. However, in the proposed network-based model the combination can be achieved by simply taking a union or intersection of the edges as needed. Furthermore, as we will show in Section 5, the analytical results obtained from the individual networks can be integrated using Boolean operations to obtain the same results that we would have obtained from the combined network. Thus we only need to solve n analytical problems and use these to obtain the results for the rest of the feature combinations. This part is demonstrated using community detection task. *Finally*, this representation as multilayered network facilitates handling of new instances as well as features. Not only can new entries be easily added via simple node, edge and/or layer addition, but the results can be updated quickly by simply combining the new result with the old ones via Boolean operations. Moreover, using link prediction algorithms, missing data can also be inferred.

To summarize, our **main contributions** are as follows:

- We propose and discuss the benefits of a multilayered network representation for data-intensive events with a large number of features captured by multiple sources (Section 3).
- We define composability rules based on Boolean operations to combine multiple features into AND, OR and NOT-composed networks and thus aid in multiple feature based analysis (Section 4).
- We introduce the concept of *self preserving* communities. We show that if the communities in the individual networks are self-preserving, then the communities in AND-composed networks can be recreated by intersecting the communities obtained from individual networks. This showcases that it is possible to infer the combined effect of features without generating the respective composed networks, thus reducing modeling complexity and improving efficiency (Section 5).
- We augment our analytical results with empirical results on a dataset of traffic accidents. We show that it is important to consider all the different subsets of the features, as each of them affects an event in a unique manner. We also empirically show that by composing the communities we can reduce the computational costs of finding communities in the AND-composed networks. (Section 6).

2 Related Work

In this section we provide an overview of the related work including work in data fusion, multilayered networks, and detecting communities in multilayered networks.

Data Fusion: The area of data fusion concerns combining data from multiple sources (including video [4]) to gain a holistic understanding of the situation. The main challenges in achieving high level information fusion [3, 5] is to link information over different collections so that user queries can be answered based on information extracted from images, videos, and correlated with other sources. Here we propose to use a multilayered network approach to fuse data associated with multiple features, of multiple types and obtained from multiple sources.

Multilayered Networks: Recently, many analytical tasks have used multilayered network [12] to handle varying interactions among the same set of entities such as co-authorship network in different conferences [6], citation network across different topics, interaction network based on

calls/bluetooth scans [9] and friendship network across different social media platforms. In each of these cases, the relationship among the entities is of the same type, and well-defined. Examples include whether people work/interact with each other, cite each other or are friends with each other. In contrast, we are interested in a class of events that are associated with features of different types, each feature providing a unique perspective. There is yet not much work on how different feature types can be combined to generate a multilayered network for representing various relationships among the same set of nodes.

Further, in order to holistically study an event, we also have to study the impact of the combinations of different layers (or features) in the network. Although, techniques based on information theory have been proposed for multilayer protein-protein interactions [8], this is only for reducing the number of redundant layers through aggregation, but not as a generalized approach for composing different layers to represent the corresponding combination of features as proposed here.

Community Detection: Community detection involves finding groups of tightly connected vertices in a network. This is a well-studied problem in network analysis, and recent work has also looked into community detection algorithms for multilayered networks [11]. Here we propose a novel approach by which communities obtained from individual layers can be easily combined to obtain communities present in the composed multilayered network. To the best of our knowledge, this technique of inferring the communities of the combined network from layers of individual communities has not been studied before.

3 Creating Multilayered Networks

In this section we describe how we create the multilayered networks from multi-source datasets. The notations introduced in this section to formalize our definitions are summarized in Table 1.

Table 1: List of notations used for defining the concepts.

N_f	Number of event features
N_I	Number of event instances
I_i	The i^{th} event instance
f^k	The k^{th} event feature
t_k	f^k type $\in \{numeric, nominal, date, time, location\}$
f_i^k	Value of I_i for f^k
$D_{t_k}(f_i^k, f_j^k)$	Distance between I_i and I_j based on f^k
τ_{f^k}	Threshold value for similarity with respect to f^k
$L(V_k, E_k)/L_k$	The k^{th} layer
V_k	Set of nodes in the k^{th} layer
E_k	Set of edges in the k^{th} layer
(u_i^k, u_j^k)	An edge between I_i and I_j in the k^{th} layer

Multi-Source Datasets: Many events are associated with multiple features (or attributes). For example, an accident scene can be described, by several features including light conditions, weather conditions, road conditions, date, etc. Therefore, each event can be described as a tuple of features. Formally, if N_I is the total number of event instances and N_f is the total number of event features, then in general the i^{th} event instance, I_i , can be represented as an N_f -tuple, shown

in equation 1, where f_i^j is the j^{th} feature's value.

$$I_i = \langle f_i^1, f_i^2, \dots, f_i^{N_f} \rangle \forall i \in \{1, N_I\} \quad (1)$$

We define *distance metric* as the measure of similarity between two event instances. The distance metric is denoted by $D_{t_k}(f_i^k, f_j^k)$ and represents the distance between the i^{th} and the j^{th} instances with respect to the k^{th} event feature, which is of type t_k . For each type of feature, multiple distance metrics are possible. Thus, the sample distance measure $D_{t_k}(f_i^k, f_j^k)$ for the different feature types considered for this paper are as follows. Note that we define the distance such that lower distance indicates higher similarity.

- *Numeric* ($t_k = \text{numeric}$): Features such as number of casualties caused by the accident and the speed limit of the road where the accident occurred, whose values correspond to integers or floating point numbers fall under this category. Equation 2 defines the distance metric as the absolute difference between the values of the features.

$$D_{t_k}(f_i^k, f_j^k) = |f_i^k - f_j^k| \quad (2)$$

- *Nominal* ($t_k = \text{nominal}$): Nominal features have a fixed discrete set of values. For example, the domain for the feature capturing road surface conditions is {dry, wet/damp, flood, snow, frost/ice, oil, mud}. For such features, Equation 3 states that the distance metric is given as 0 if there is an exact match and undefined (denoted as ϕ) if they do not match.

$$D_{t_k}(f_i^k, f_j^k) = \begin{cases} 0 & \text{if } f_i^k = f_j^k \\ \phi & \text{if } f_i^k \neq f_j^k \end{cases} \quad (3)$$

- *Date* ($t_k = \text{date}$): Certain features will depict the date of occurrence of the event instance. The distance measure is the number of days between the occurrences of two instances.

$$D_{t_k}(f_i^k, f_j^k) = \text{daysBetween}(f_i^k, f_j^k) \quad (4)$$

- *Time* ($t_k = \text{time}$): This feature gives the exact time of the occurrence in hours (HH), minutes (MM) and seconds. To compute the distance metric we divide the day into 48 intervals of 30 minutes each from [0000-0030) to [2330-0000). We assume that two events taking place around the same time interval may be similar in nature, even if they happen on two different dates. For example, a set of accidents may be similar because they occur during the evening rush hour on any of the weekdays. Thus, Equation 5 states that for a time based feature, the number of 30 minute intervals between the occurrences of two given instances, will be used as the distance measure.

$$D_{t_k}(f_i^k, f_j^k) = ||[2 * f_i^{k_{HH}} + 1 + \lfloor f_i^{k_{MM}} / 30 \rfloor] - [2 * f_j^{k_{HH}} + 1 + \lfloor f_j^{k_{MM}} / 30 \rfloor]|| \quad (5)$$

- *Geographical Location* ($t_k = \text{location}$): The geographical location of an event's occurrence is given by its latitude value (LAT) and longitude value (LONG). We use the Haversine formula ([1]) that calculates great-circle distance between any two points on the earth's spherical surface to define the distance metric in Equation 6 for the location based features, considering R to be the radius of the earth.

$$D_{t_k}(f_i^k, f_j^k) = 2R \arcsin * (\sqrt{(\sin^2(\frac{f_i^{k_{LAT}} - f_j^{k_{LAT}}}{2}) + \cos(f_i^{k_{LAT}}) * \cos(f_j^{k_{LAT}}) * \sin^2(\frac{f_i^{k_{LONG}} - f_j^{k_{LONG}}}{2}))}) \quad (6)$$

In addition to the types listed above, other types such as videos, images, audio files, tweets, SMS etc. can also enhance the description of the event. Currently we are not considering these types for this paper.

Creating the Multilayered Network: Based on the distance metric we now represent the dataset as a multilayered network (or graph). For a given feature, we say that a pair of event instances are similar if their distance metric is below a specified threshold. We create a separate network for each feature. The instances are represented as vertices in the network. Two vertices are connected in a network, if for that corresponding feature, they are similar. Therefore to create a layer of the network based on a specific feature, we need the following information:

- A set, I , of all the event instances such that $I = \{I_1, I_2, \dots, I_{N_I}\}$
- The type of the i^{th} feature, t_i . For the current paper, we have considered $t_i \in \{numeric, nominal, date, time, location\}$.
- The metric, $D_{t_i}(f_m^i, f_n^i)$, defined to calculate the distance between any two event instances, I_m and I_n .
- A specified threshold value, τ_{fi} , that dictates the similarity between any two instances with respect to the i^{th} feature.

Formally, in the i^{th} layer, the j^{th} instance, I_j , will be depicted by the j^{th} vertex, u_j^i . The presence of an undirected and unweighted edge in this layer, (u_j^i, u_k^i) , will depict that the j^{th} and the k^{th} instances are similar to each other with respect to the i^{th} feature. Each network layer can be uniquely defined by the feature it represents, and will be denoted as $L(V_i, E_i)$ or L_i . Note that every layer will have the same set of nodes, but different set of edges.

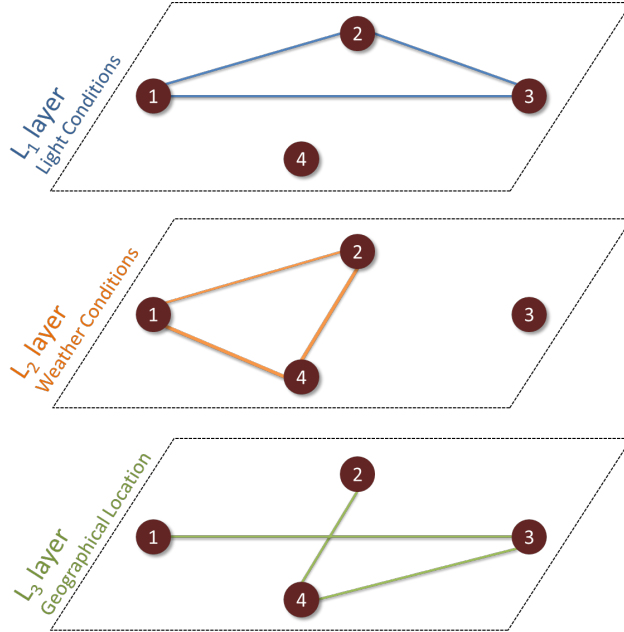


Figure 1: Snapshot of Multilayered network for the accident event

Figure 1 shows a multilayered network for four accident instances, denoted by four nodes numbered from 1 to 4. Similarity among the accidents is considered with respect to two nominal features - Light Conditions and Weather Conditions and one location based feature - (Latitude, Longitude),

with the threshold value for distance metric being 2 miles. Note that the connectivity of each layer in the network is different, highlighting the unique perspective of each feature. For instance, accident2 and accident3 had the same light conditions when they occurred, but didn't share the weather conditions. This small snapshot shows that every feature tells a different perspective about the relationship among the accidents, thus supporting the relevance of analyzing any event in a perspective-wise manner.

Representing multi-featured datasets using multilayered networks provides the following benefits:

- *Ease of handling the dataset incrementally.* The multilayered network representation makes it easy to add or delete new entries and features into the data set. This is because each layer is generated independently from other layers and hence only the affected layer has to be changed through the addition or deletion of nodes and/or edges. As we will see in the next section, even combining the features is made more effective due to the multilayered approach.
- *Identifying importance of features on the central event.* The multilayered framework allows us to analyze the contribution of individual or combined features. The importance of a feature can be measured by factors such as the edge density of the network, the number of connected components and the community structure. These measurements can help us order the features in terms of their importance.
- *Determining the strength of a relationships.* The network-based representation allows us to easily identify the strength of the relationships between event instances. For example, the instances that have an edge between them across multiple layers are more strongly related than if they have an edge in only one layer.
- *Inferring feature dependencies and missing instance-instance relationships.* Every layer has the same set of nodes but different set of edges. Thus, based on the edge connectivity a correlation can be identified among the features. In case of missing feature values, these inferred correlations among the features will aid the link prediction algorithms to infer the missing relationships.
- *Efficient computation of the effect of multiple features.* Combining different subsets of n features requires us to solve 2^n separate problems. However, the network-based representation allows us to easily combine different layers in any arbitrary manner using Boolean operations (Section 4). Moreover, results of an analytical task for a combined network can be obtained by only using the results from the individual layers. This aspect is discussed in more detail with respect to the community detection task in Section 5.

4 Layer Composition Through Boolean Operations

Each layer in the network provides information of how a single feature effects the event instances. However, it is extremely pivotal to study the effect of the combination of features, as each combination presents a new perspective. For a set of n features, a total of 2^n different feature combinations are possible.

For example, for an accident dataset with two features, light and weather, the combinations can be generated in the following 4 ways: i) both light and weather, ii) either light or weather, iii) light and not weather (only light) *or* iv) weather and not light (only weather).

However, creating a network with multiple features leads to an additional challenge of how to compute the distance metric for a combination of features. To address this challenge we propose

the use of fundamental Boolean algebraic operators. Therefore, with respect to the multilayered network, the distance measure criterion based on k features should correspond to the k -layer combination scheme - $(L_{i1} \theta L_{i2} \theta \dots \theta L_{ik})$, where θ represents the type of boolean operator, i.e. AND (Section 4.1), OR (Section 4.2) and NOT (Section 4.3).

4.1 AND Composition

The AND composition over a set of layers includes an edge only if it occurs in *all* the layers. This indicates that the pair of event instances connected by the edges satisfies the threshold parameter for all the required features.

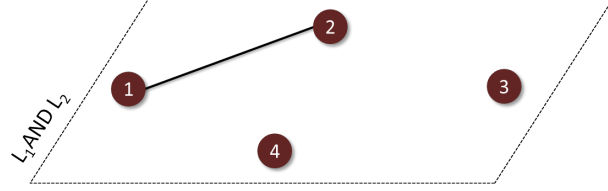


Figure 2: 2-layer AND composition applied on the Light layer and Weather layer present in Figure 1

In Figure 1, for L_1 and L_2 , the set of edges E_1 and E_2 depict the accident pairs that are similar based on light and weather, respectively. Figure 2 depicts the $L_1 \text{ AND } L_2$ composition, that contains only edges that are present in both E_1 and E_2 .

Formally, the AND composition of two layers, $L(V_i, E_i)$, $L(V_j, E_j)$, will produce the composed layer $L(V_{iANDj}, E_{iANDj})$. A representative vertex, u_m^{iANDj} , is added to the set of vertices, V_{iANDj} , for each event instance I_m . For any event instance pair, I_m and I_n , if an edge exists between their representative vertices in both layer L_i and layer L_j , then an edge, $(u_m^{iANDj}, u_n^{iANDj})$ becomes a part of the set of edges, E_{iANDj} . The steps for 2-layer AND composition are given in Algorithm 1.

Algorithm 1 Algorithm for AND composition

Require: $\langle L(V_i, E_i), L(V_j, E_j) \rangle$, $V_{iANDj} = \emptyset$, $E_{iANDj} = \emptyset$

- 1: **for all** $u_m^i \in V_i$ **do**
 - 2: $V_{iANDj} \leftarrow V_{iANDj} \cup u_m^{iANDj}$
 - 3: **end for**
 - 4: **for all** $u_m^{iANDj}, u_n^{iANDj} \in V_{iANDj}, m > n$ **do**
 - 5: **if** $(u_m^i, u_n^i) \in E_i$ **AND** $(u_m^j, u_n^j) \in E_j$ **then**
 - 6: $E_{iANDj} \leftarrow E_{iANDj} \cup (u_m^{iANDj}, u_n^{iANDj})$
 - 7: **end if**
 - 8: **end for**
-

A k -layer AND composed network $(L_{AND_{j=1}^k(ij)})$ indicates that when combining layers $(L_{i1} \text{ AND } L_{i2}) \text{ AND } L_{i3} \dots \text{ AND } L_{ik}$, a pair of event instances will have an edge between them if $D_{t_{ij}}(f_m^{ij}, f_n^{ij}) \leq \tau_{f^{ij}}$, for *every* $j \in [1, k]$. Equation 7 shows that the number of edges in an AND composed layer will be bounded by the number of edges in layer with the lowest number of connections, since the composition is formed by an *intersection of edges*.

$$0 \leq |E_{AND_{j=1}^k(ij)}| \leq \min_{j \in [1, k]} |E_{ij}| \quad (7)$$

4.2 OR Composition

The OR composition over a set of layers includes an edge if it occurs in any one of the constituent layers. This indicates that the pair of event instances connected by the edges satisfies the threshold parameters for at least one of the features.

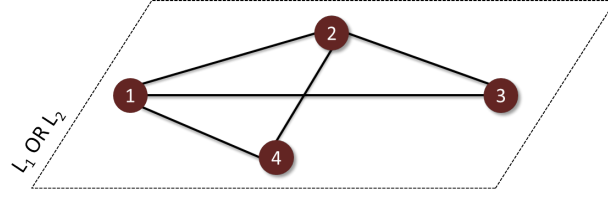


Figure 3: 2-layer OR composition applied on the Light layer and Weather layer present in Figure 1

For example, for the light and weather based layers in Figure 1, the expected result for L_1 OR L_2 composition is a set, which contains edges present in either E_1 or E_2 , or both, as shown in Figure 3.

The methodology to perform OR composition is similar to the AND Composition and is given by Algorithm 2. For the same individual layers considered in Section 4.1, the OR composed layer will be $L(V_{iORj}, E_{iORj})$. For every event instance I_m , the set V_{iORj} will contain its representative vertex, u_m^{iORj} . An edge, (u_m^{iORj}, u_n^{iORj}) will be introduced in this composed layer, if the representative vertices of I_m and I_n , have an edge between them in either layer L_i or layer L_j .

Algorithm 2 Algorithm for OR composition

Require: $\langle L(V_i, E_i), L(V_j, E_j) \rangle, V_{iORj} = \emptyset, E_{iORj} = \emptyset$

- 1: **for all** $u_m^i \in V_i$ **do**
 - 2: $V_{iORj} \leftarrow V_{iORj} \cup u_m^{iORj}$
 - 3: **end for**
 - 4: **for all** $u_m^{iORj}, u_n^{iORj} \in V_{iORj}, m > n$ **do**
 - 5: **if** $(u_m^i, u_n^i) \in E_i$ **OR** $(u_m^j, u_n^j) \in E_j$ **then**
 - 6: $E_{iORj} \leftarrow E_{iORj} \cup (u_m^{iORj}, u_n^{iORj})$
 - 7: **end if**
 - 8: **end for**
-

A k -layer OR composed network ($L_{OR_{j=1}^k(ij)}$) indicates that when combining layers (L_{i1} OR L_{i2} OR L_{i3}) ... OR L_{ik}), a pair of event instances will have an edge between them if $D_{t_{ij}}(f_m^{ij}, f_n^{ij}) \leq \tau_{f^{ij}}$, for *at least one* $j \in [1, k]$. Since the composition is formed by an *union of edges*, the number of edges in an OR composed layer will be bounded by the total number of edges in all the constituent layers, which is shown in Equation 8.

$$\max_{\forall j \in [1, k]} |E_{ij}| \leq |E_{OR_{j=1}^k(ij)}| \leq \frac{N_f(N_f - 1)}{2} \quad (8)$$

4.3 NOT Composition

The NOT composition models the complement of a feature. Thus, NOT composition for the k^{th} layer will generate a network where the existence of an edge will imply that the accident pair does not satisfy the threshold parameter for the k^{th} feature. For example, for the light based layer in Figure 1, the expected result for *NOT* L_1 composition is shown in Figure 4.

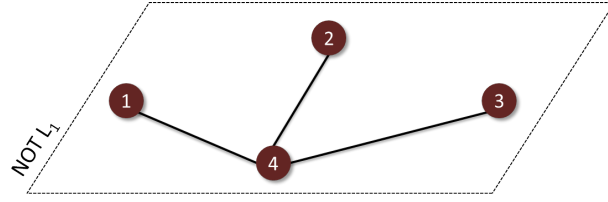


Figure 4: NOT composition of the Light layer present in Figure 1

It can be observed in Algorithm 3 that unlike the AND and OR composition, the NOT composition is applied on a single layer. The NOT of the k^{th} layer will be a new layer $L(V_{k'}, E_{k'})$, where $V_{k'}$ contains a representative vertex, $u_m^{k'}$, for each event instance, I_m . For any two event instances, I_m and I_n , an edge $(u_m^{k'}, u_n^{k'})$ is introduced if the representative nodes of these instances do not contain an edge between them in the original layer.

Algorithm 3 Algorithm for NOT composition

Require: $L(V_k, E_k)$, $V_{k'} = \emptyset$, $E_{k'} = \emptyset$

- 1: **for all** $u_m^k \in V_k$ **do**
 - 2: $V_{k'} \leftarrow V_{k'} \cup u_m^{k'}$
 - 3: **end for**
 - 4: **for all** $u_m^{k'}, u_n^{k'} \in V_{k'}, m > n$ **do**
 - 5: **if** $(u_m^k, u_n^k) \notin E_k$ **then**
 - 6: $E_{k'} \leftarrow E_{k'} \cup (u_m^{k'}, u_n^{k'})$
 - 7: **end if**
 - 8: **end for**
-

The set of edges for this layer, $E_{k'}$, will correspond to the *complement of the set of edges in the k^{th} layer*. Therefore, for any two instances, the existence of an edge in $L_{k'}$ depicts that the condition $D_{t_k}(f_m^k, f_n^k) > \tau_{fk}$ is satisfied. From Equation 9 it can be concluded that the density of the NOT composed layer will be inversely proportional to the density of the original layer.

$$|E_{k'}| = \frac{N_f(N_f - 1)}{2} - |E_k| \quad (9)$$

Complex Composition of Layers: Primitive Boolean operations can be used to create more complex compositions, as shown in Table 2 and Figure 5. Since these layer compositions are based on Boolean algebra, they will also obey the associative, commutative, distributive and De Morgan's laws, as displayed in Table 3. Using these properties, any complex layer composition of layers can be expressed using the defined AND, OR, and NOT operations.

Table 2: Complex Layer Compositions

$L_1 \text{ NAND } L_2$	$NOT (L_1 \text{ AND } L_2)$
$L_1 \text{ NOR } L_2$	$NOT (L_1 \text{ OR } L_2)$
$L_1 \text{ XOR } L_2$	$(L_1 \text{ AND } (NOT L_2)) \text{ OR } ((NOT L_1) \text{ AND } L_2)$

This section showed how the individual layers in the multi-layered framework can be combined in various ways in order to produce new layers, each presenting an interesting perspective of looking into the relationship among the event instances. In this way, this architecture allows anyone to analyse the impact of multiple features on the central event.

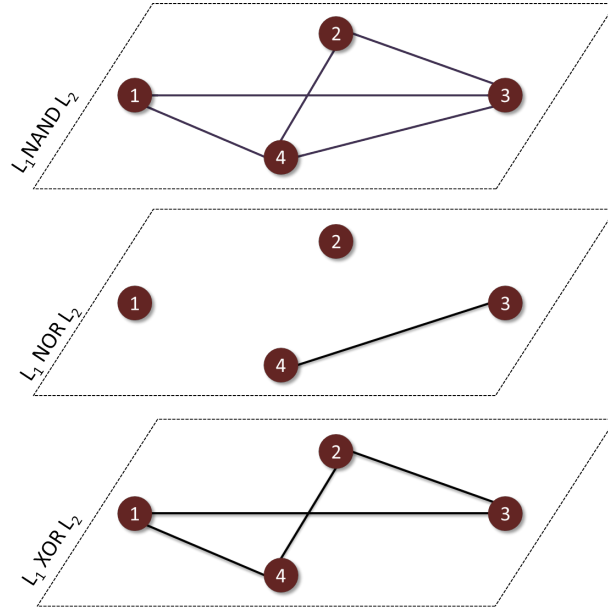


Figure 5: Complex compositions for the layers shown in Figure 1

Table 3: Layer Composition Properties

Commutativity	$L_i \theta_1 L_j \equiv L_j \theta_1 L_i$
Associativity	$(L_i \theta_1 L_j) \theta_1 L_k \equiv L_i \theta_1 (L_j \theta_1 L_k)$
Distributivity	$L_i \theta_1 (L_j \theta_2 L_k) \equiv (L_i \theta_1 L_j) \theta_2 (L_i \theta_1 L_k)$
De Morgan's	$NOT (L_i AND L_j) \equiv (NOT)L_i OR (NOT)L_j$ $NOT (L_i OR L_j) \equiv (NOT)L_i AND (NOT)L_j$
	where, L_i, L_j, L_k : basic/composed layers $\theta_1, \theta_2 \in \{AND, OR\}$

5 Combining Analytical Results Using Boolean Functions

For a given dataset, one of the primary tasks that we are interested in this paper is to show that analytical results with respect to a combination of features can be inferred by just using the results obtained with individual features. To illustrate this, we chose the analytical task as the clustering of event instances (accidents in our example) based on the single or combined set of features. In the network context, this is equivalent to identifying groups of tightly connected vertices or communities [10, 13].

Although we presented an elegant method for combining individual layers of networks using Boolean operations, we still have to find the communities in these different combined networks. Thus we have to solve 2^n separate community detection problems.

In this section, we analytically show that if communities follow certain characteristics then we can reproduce the communities of the composed networks. This reduces the memory requirements since we do not have to load each of the separate composed networks into memory and also reduces computational time because we can recreate the communities using simple Boolean operations, rather than expensive community detection methods.

Recreating Communities in AND-composed Networks: We first introduce the concept

of *self preserving* communities. A community is self preserving if the vertices in the community are so strongly connected such that even if only a subset of connected vertices remain in a community, they will form a smaller community rather than joining an existing larger community.

Formally, consider a network G , that has a community whose vertices are given by the set C_v . Now consider the network induced by a subset of vertices $C_v^S \in C_v$ and all other vertices that are not in C_v . If the vertices in C_v^S form a community by themselves, for any subset C_v^S of C_v , where $\|C_v^S\| \geq 3$ and the vertices in C_v^S are connected, then community C_v is self preserving.

Now, consider two networks $G1$ and $G2$ that have the same set of vertices, but different set of edges. Moreover, both networks have only self-preserving communities. Now consider the AND-composition of $G1$ and $G2$, G_{AND} . Only edges that are in both $G1$ and $G2$ will be in the AND-composed network. Therefore the communities formed in the AND-composed network will be based on a subset of edges from $G1$ and $G2$. Since both $G1$ and $G2$ have self preserving communities, therefore the communities formed in G_{AND} will be formed subsets of the communities in $G1$ and $G2$. Most importantly, due to the self preserving nature, no new grouping of vertices will be formed in G_{AND} . Therefore we can reconstruct the communities in G_{AND} by simply taking the intersection of the communities of $G1$ and $G2$.

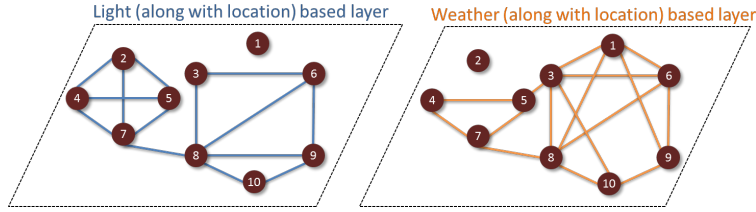


Figure 6: The layers (along with location) generated for a random set of accident instances

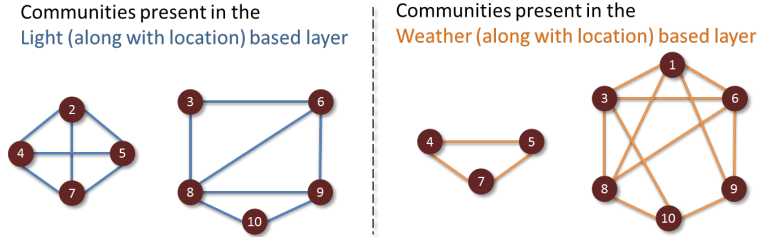


Figure 7: Actual Communities generated for the layers in Figure 6

An example of such reconstruction is given in Figures 6- 9. Figure 6 shows two layers of networks and Figure 7 their corresponding communities, all of which are self-preserving. Figure 8 shows the AND-composed network and the resultant communities. Figure 9 shows that for this toy example, we can indeed reconstruct the communities for the AND-composed networks by taking the intersection of the communities from the two separate networks.

6 Experimental Results

In this section we present our experimental results on composing networks with combined features and recreating the communities in these composed networks. Specifically, we i) construct user-defined individual layers, ii) perform Boolean compositions of the generated individual layers and iii) validate that the communities obtained by intersection of the individual layers are the same as the communities obtained by the composed layer.

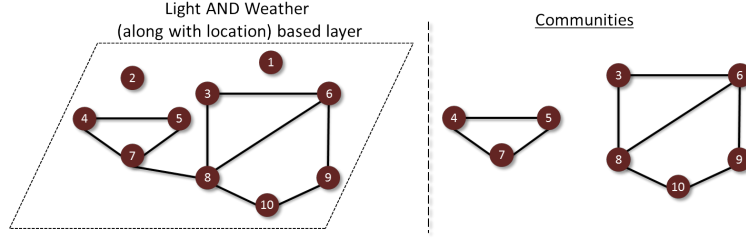


Figure 8: Actual Communities for the Light AND Weather (along with location) based layer

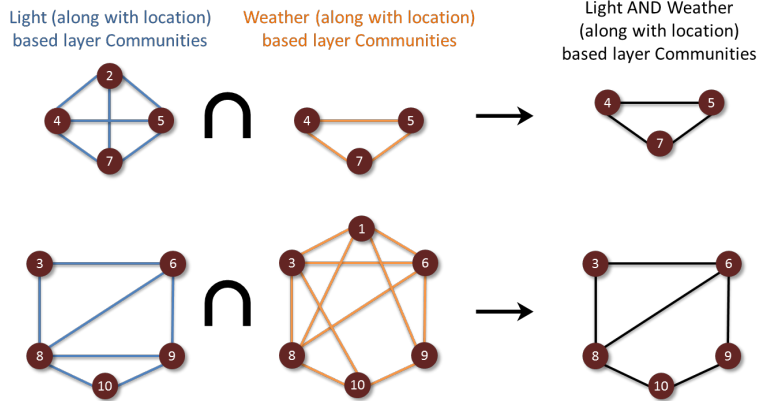


Figure 9: Pairwise intersection of the communities in the individual layers *recreates* the communities of the AND composed layer

We use a dataset of road accidents that occurred in the United Kingdom in the year 2014 [2]. Out of a total of 32 attributes captured for each accident, we use three nominal features - light conditions with domain as $\{\text{daylight}, \text{darkness: lights lit}, \text{darkness: lights unlit}, \text{darkness: no lighting}, \text{darkness: lighting unknown}\}$, weather conditions with domain as $\{\text{fine} + \text{no high winds}, \text{raining} + \text{no high winds}, \text{snowing} + \text{no high winds}, \text{fine} + \text{high winds}, \text{raining} + \text{high winds}, \text{snowing} + \text{high winds}, \text{fog or mist}, \text{other}\}$ and road surface conditions with domain as $\{\text{dry}, \text{wet or damp}, \text{snow}, \text{frost or ice}, \text{flood}, \text{oil or diesel}, \text{mud}\}$ for the first three individual layers (L_1 , L_2 and L_3). The latitude and longitude values of accident location were grouped to form the geographical location based fourth layer, L_4 and time was the fifth layer L_5 .

Our codes were implemented in C++ and were executed on a Linux based machine with 4 GB RAM, 500GB of local disk space and installed with UBUNTU 13.10. We used Infomap [7] to detect communities in the networks, with a setting which assigns any node to at most one community.

Generating the Layers per Feature: Three layers of our network, light, weather and road conditions are of nominal type, therefore an edge is added if the values match exactly. For layers L_4 and L_5 the appropriate threshold has to be determined. There is a trade-off here, because too low a threshold can lead to loss of information, and too high a threshold leads to a dense network that is expensive to analyze.

Identifying appropriate thresholds: To identify the appropriate threshold, we plotted different thresholds for distance (L_4) and time (L_5) layers versus the density of the layer at that threshold. Figure 10 shows that the change in layer density, peaks in the interval 10-12 miles for the distance layer and in the interval 3.5-4 for the time layer, which is divided into 30 minute slots. Based on this information we selected the threshold for the distance at 10 miles and threshold for time at 1.5 hours.

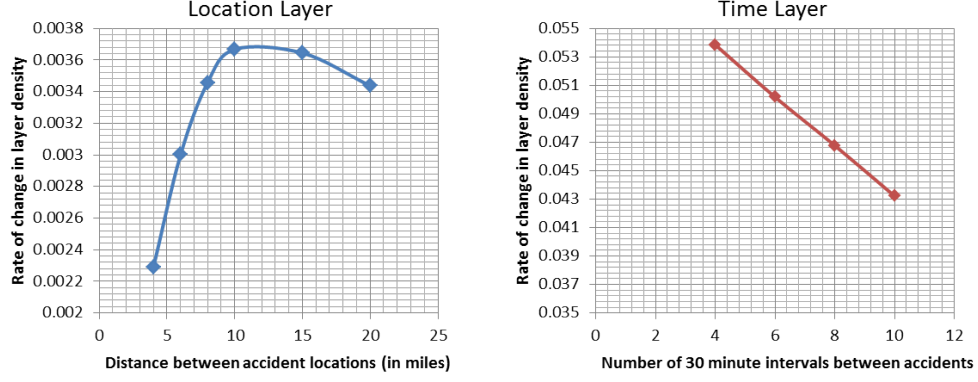


Figure 10: Variation in rate of change in layer density with threshold value

Density of basic and composed layers: In Figure 11, we show the densities of the individual nominal layers (Light, Weather), their different composed layers (AND and OR) and the complement graph for the Light layer (NOT) for a set of 1000 accidents from the dataset. The density of the AND-composed layer ($L_1 \text{ AND } L_2$) will have an upper bound of the minimum density between L_1 and L_2 , because it is formed of the intersection of the edges. Similarly, the union of all edges causes the density of the OR-composed layer ($L_1 \text{ OR } L_2$) to have a lower bound of the maximum density between L_1 and L_2 .

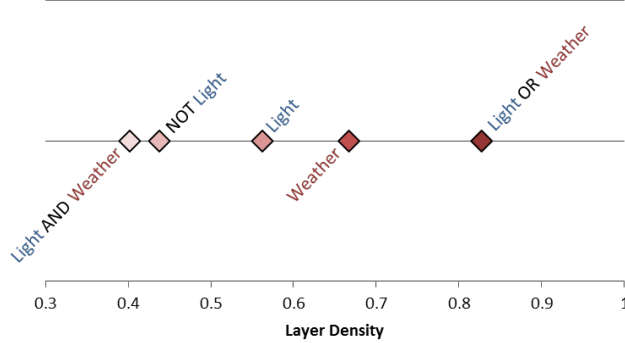


Figure 11: Distribution of densities for individual and composed layers for the accident event

For our experiments we AND-composed each of the nominal layers with the distance and time layers to ensure that we are considering accidents within the same distance radius and same time interval. Thus when we refer to the Light layer we mean that it is Light AND Distance AND Time. We refer to the Weather and Road Condition layers similarly.

Communities in the Individual and Composed Layers: We now find the communities in the individual and composed layers to identify groups of accidents that are influenced by a similar set of features. In Figure 12 we plot a random set of a few accidents and their respective communities in the Light, Weather and Road layers. The X-axis shows the Id of the accident and the Y-axis the community to which the accident belongs. The squares, triangles and circles, indicate the communities obtained from the Light, Weather and Road layers respectively.

As can be seen from the figure, there are several accidents that are assigned to the same community by multiple layers. But there are certain accidents like accident number 23 and 24 that are assigned to the same community as per the Road layer, but to different communities as per the Light and Weather layers. The **main takeaway is that there are accidents that are**

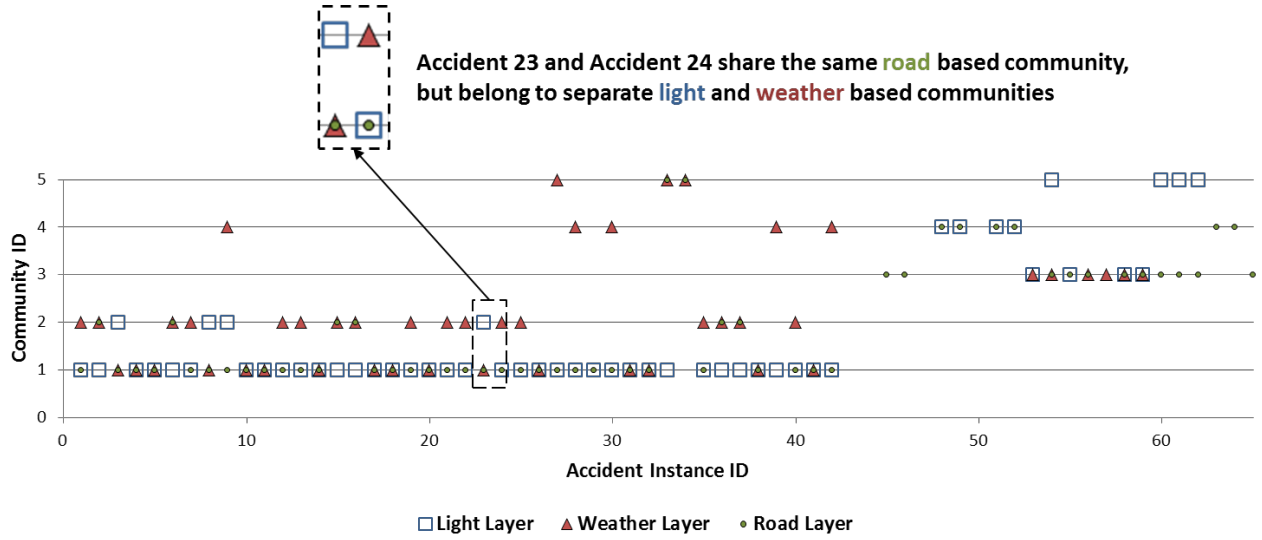


Figure 12: Minimal overlap among layer-wise communities for a snapshot of random accident instances

influenced by different subgroups of features.

We show the breakup of how 1000 accidents are grouped by the various individual and composed layers in Figure 13. The pie-chart shows that 60% of the accidents were grouped based on *all the features*. 5% of the accidents were not in any community. Therefore, a multilayer analysis of all features will lose information of the 35% accidents that belonged to some community in other composed and individual layers. This highlights that **it is equally important to analyze the individual layers and their various compositions**.

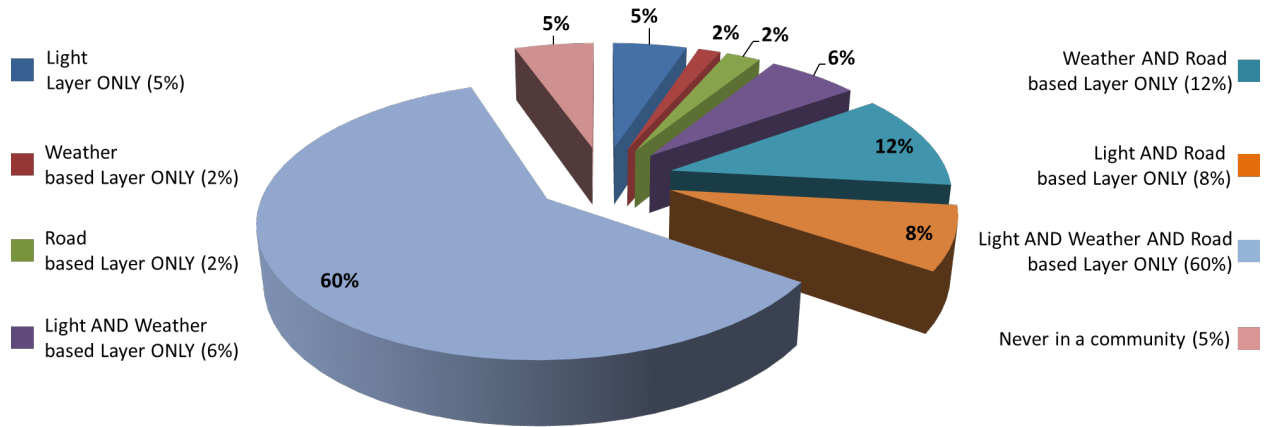


Figure 13: Percentage of instances that belong to some community with respect to individual or combination of features/layers (for 1000 random accident instances)

Recreation of communities in the AND-composed networks: As discussed earlier, computing communities from each of the composed networks is an expensive task. Here we show that we can successfully recreate the communities of the AND-composed layers, thus reducing the space to store the AND-composed layers and also the time.

We noticed that all the communities in the Light, Weather and Road layers were self-preserving. Therefore we can recreate the communities in the AND-composed layers by simply intersecting the communities in the individual layers.

Figure 14 shows the similarity between the communities created from the AND-composed networks (Light AND Weather, Light AND Road, Weather AND Road, and, Light AND Weather AND Road) and the communities recreated by intersecting the communities of the individual layers for 3000 accident instances. The similarity between the communities was computed using the Jaccard Index (J). For two sets A and B , $J_{A,B} = \frac{A \cap B}{A \cup B}$. Thus a Jaccard value of 1 means that the two sets are identical. As can be seen from the sub-figures that the Jaccard value was 1 for the 5 largest communities for each of the AND-composed networks. We observed exactly the same results ($J = 1$) when testing on smaller datasets of 1000 and 2000 accident sets. This empirically validates that **the communities in AND-composed networks can be successfully recreated by intersecting the communities in the individual networks.**

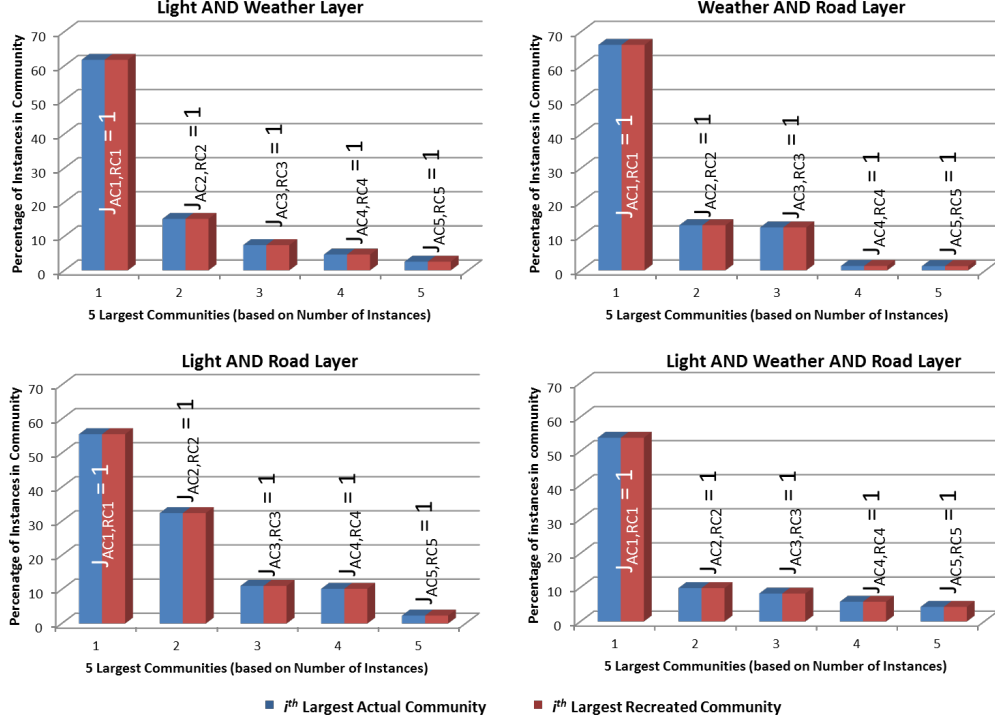


Figure 14: Comparison of the Jaccard Index ($J_{ACi,RCi}$) between the i^{th} largest actual community and the i^{th} largest recreated community, for various AND-compositions of Light, Weather and Road layers, for 3000 accidents.

Time to re-create the communities. Figure 15 compares the time to re-create the communities versus the time to generate them in the AND-composed networks on the 3000 accident dataset. To generate the communities in the individual layers it took **7.406 seconds**, **8.504 seconds** and **7.08 seconds**, for Light, Weather and Road Layers, respectively. After that it took **5.372 seconds**, **5.072 seconds**, **5.032 seconds** and **4.96 seconds** to perform the intersection of layer-wise communities to recreate the communities for Light AND Weather AND Road, Light AND Weather, Weather AND Road and Light AND Road composed layers, respectively.

In comparison it took **22.691 seconds**, **13.265 seconds**, **14.465 seconds** and **12.08 seconds** to create the above mentioned AND-composed layers and **3.992 seconds**, **5.271 seconds**, **6.122 seconds** and **4.438 seconds** to obtain the communities for them, respectively. Therefore, the recreation method was about 47% faster, a total of **43.426 seconds** compared to a total of **82.324 seconds**. This is likely to improve further as the number of features increases.

These experimental results highlight that multilayered network is an effective tool for studying

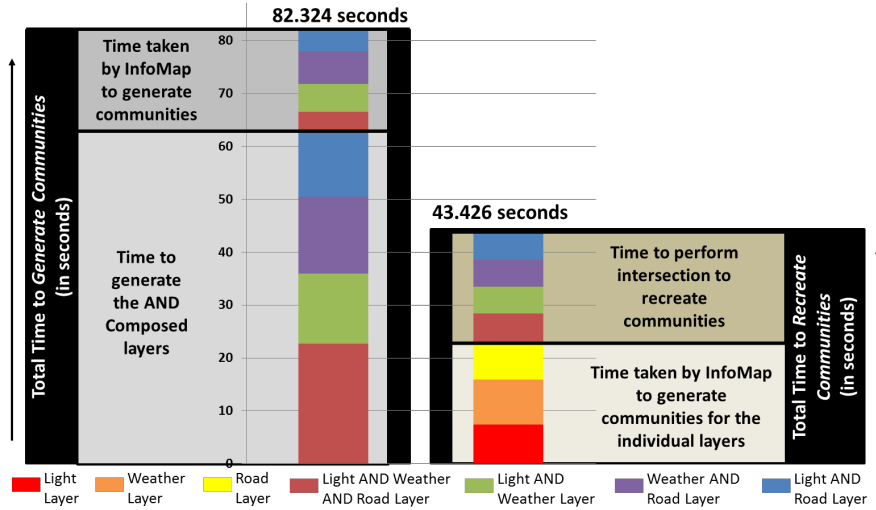


Figure 15: Comparison of time between generating and recreating the communities for AND-Composed Layers

events associated with multiple data. They also show that in order to have a holistic understanding of the central event perspective-wise analysis is the key, that is we need to study the effect of all combinations of the features. Finally we show that recreating the communities from individual layers can reduce the computational costs of the analysis.

7 Conclusion and Future Extensions

This paper proposes a novel approach to model and analyze data fusion problems. This paper makes a case for multilayered analysis approach for multi-source, data fusion problems, its advantages, and composability aspects to improve modeling and computation aspects. Initial experimental results on real-world datasets have been very encouraging and empirically establish composability.

As future work, we plan on extending this work by introducing weighted and directed edges, modifying the composition schemes with respect to such type of edges, handling other types of features and distance metrics and come up with a generalized formulation for inferring communities for k-level composed layers/features based on single feature based communities, along with the theoretical analysis for this method’s prediction accuracy.

References

- [1] Haversine formula. https://en.wikipedia.org/wiki/haversine_formula.
- [2] Road safety - accidents 2014. <https://data.gov.uk/dataset/road-accidents-safety-data/resource/1ae84544-6b06-425d-ad62-c85716a80022>.
- [3] E. Blasch, J. Llinas, D. Lambert, P. Valin, S. Das, C. Chong, M. Kokar, and E. Shahbazian. High level information fusion developments, issues, and grand challenges: Fusion 2010 panel discussion. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8. IEEE, 2010.

- [4] E. Blasch, J. Nagy, A. Aved, W. Pottenger, M. Schneider, R. Hammoud, E. Jones, A. Basharat, A. Hoogs, G. Chen, et al. Context aided video-to-text information fusion. In *Intl Conf. on Information Fusion*, 2014.
- [5] E. P. Blasch, D. A. Lambert, P. Valin, M. M. Kokar, J. Llinas, S. Das, C. Chong, and E. Shahbazian. High level information fusion (hlif): Survey of models, issues, and grand challenges. *Aerospace and Electronic Systems Magazine, IEEE*, 27(9):4–20, 2012.
- [6] B. Boden, S. Gnnemann, H. Hoffmann, and T. Seidl. Mining coherent subgraphs in multi-layer graphs with edge labels. In *Proc. of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD 2012), Beijing, China*, pages 1258–1266, 2012.
- [7] L. Bohlin, D. Edler, A. Lancichinei, and M. Rosvall. Community detection and visualization of networks with the map equation framework. 2014.
- [8] M. D. Domenico, V. Nicosia, A. Arenas, and V. Latora. Layer aggregation and reducibility of multilayer interconnected networks. *CoRR*, abs/1405.0425, 2014.
- [9] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov. Clustering with multi-layer graphs: A spectral perspective. *CoRR*, abs/1106.2233, 2011.
- [10] S. Fortunato and C. Castellano. Community structure in graphs. In *Encyclopedia of Complexity and Systems Science*, pages 1141–1163. 2009.
- [11] J. Kim and J. Lee. Community detection in multi-layer graphs: A survey. *SIGMOD Record*, 44(3):37–48, 2015.
- [12] M. Kivelä, A. Arenas, M. Barthélemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *CoRR*, abs/1309.7233, 2013.
- [13] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review*, E 69(026113), 2004.